Project 0 Requirements

* *Color code: green highlights are learned topics, blue highlights are the topics implemented into project code*

**Description** Leveraging Java 8, create a console based application with a multi-tier user system. You may choose the actual use case for your application as long as it meets all the requirements defined below.

Possible examples include:

* Banking application
* Employee management system
* Collectable card game tracker
* Purchase ordering system

**Requirements**

* Technologies:
  + Java 8
    - Scanner (Console inputs)
    - JDBC (Database Connectivity)
    - JUnit (Unit Testing)
    - Logback (Logging)
  + PostgreSQL
  + Maven (Dependency Management)
* Functionalities:
* Multi-tier Users - There must be at least three different types of user accounts each different functionality access.
* Basic mathematic manipulation of the fields of an entity object.
* User login with passwords and logout.
  + Passwords may not be stored in their raw form in the database. They must be encrypted somehow.
  + Stretch Goal: User registration.
* Validation for invalid inputs.
* Reasonable Logging of user interactions with the application.
* Design:
  + Data persistence in a SQL database normalized to 3rd normal form.
    - Must contain at least three entity tables. (Join tables do not count.)
    - Must contain at least two relationships between tables.
      * Stretch Goal: Create a Many to Many relationship between tables.
    - Must contain at least one trigger.
  + Three layered architecture of the Java application.
    - Reasonable test coverage of the service layer.
      * Stretch Goal: Implement Test Driven Development.
  + DAO design pattern.
* Other:
  + A SQL Script file that can be used to create a clone of your database structures.
  + Stretch Goal: A slide show presentation introducing your application before demoing it.

**Example Bank Application Requirements**

* Customers of the bank should be able to register with a username and password, and apply to open an account.
  + Stretch Goal: Customers should be able to apply for joint accounts
* Once the account is open, customers should be able to withdraw, deposit, and transfer funds between accounts
  + All basic validation should be done, such as trying to input negative amounts, overdrawing from accounts etc.
* Employees of the bank should be able to view all of their customers information. This includes:
  + Account information
  + Account balances
  + Personal information
* Employees should be able to approve/deny open applications for accounts
* Bank admins should be able to view and edit all accounts. This includes:
  + Approving/denying accounts
  + withdrawing, depositing, transferring from all accounts
  + canceling accounts

**Evaluation**

The project will be evaluated out of 100 points split between three main catagories: 70 points for the functionality and design of your project, 25 points for the presentation of your project during the project showcase, and 5 points for the use of proper Scrum practices. The evaluation will be further subdivided as follows:

**Project Score**(70 pts):

* 15 pts: Ability to persist meaningful data in the database and then retrieve it and display it to the console.
* 10 pts: Ability for the application to logically address the use-case for which it was designed.
* 10 pts: Ability to mathematically manipulate values stored in an entity object.
* 10 pts: Proper database schema achieving 3rd normal form. (E.g. Accounts have a proper relationship to their owning user.)
* 10 pts: Login, register, update and logout functionality for users.
* 5 pts: Proper use of DAO design pattern.
* 5 pts: Different user roles with different levels of access implemented correctly.
* 5 pts: Reasonable test coverage of the service layer and proper logging.

**Presentation Score**(25 pts):

* 15 pts: Clear, concise, logical and professional communication during the project presentation.
* 10 pts: Ability to communicate clear answers to fully address questions asked about the project.

**Agile Score**(5 pts):

* 5 pts: daily participation in standup and notes recorded in the squad wiki.

**Frequently Asked Questions**

1. When is the project due?

A: The project will be due no earlier then Wednesday of Week 3. The exact due date will be set by QC as they schedule project 0 presentations.

1. Can I have an extension?

A: No. While you are encouraged to continue to work on your projects past the date of the project presentation for your own learning benefit, extensions can not be accommodated. If for some reason you can not be present at the time of project presentations you will need to schedule a time BEFORE the due date to present your project.

1. Is there a code freeze?

A: It is recommended that you institute your own code freeze at least a day before the project presentations. However, this is a recommendation only; it will not be enforced. NOTE: The code that will be evaluated by your trainer will be the code you last pushed to your repository BEFORE the time set for project presentations. Code submitted while presentations are on-going or after will not be evaluated.

1. What happens if I break my project that was mostly working right before the due date?

A: As you should have been regularly pushing code to your repository you should be able to roll back to previously working version. If you have not regularly pushed your code and do not have a working commit to return to you will need to present the state of your application in its current form.

1. Who will be evaluating the project?

A: Your trainer will be the one providing the full evaluation of your projects. However, the QC team will also be present at presentations to ask questions about your project and consult with your trainer.

1. Are we allowed to collaborate with others on our projects?

A: Collaboration is encouraged. Hopefully you will work together to solve the problems presented in this project. However, you should still be ultimately designing the project yourself. Straight copy/pasting of another person's code is considered plagiarism. NOTE: Code provided in demos by your trainer is not subject to plagiarism concerns. Feel free to copy/paste and edit such code as necessary to suit your projects.

1. What is a passing score?

A: 75% is the base expected passing score.

1. What are stretch goals?

A: Stretch goals are extensions to the project that if completed will grant bonus points. However, stretch goals will only be considered if all the required functionality (i.e. requirements that are not stretch goals) are completed.

1. If I fail the project will I be released from training?

A: The project is a major component of the determination process for continued participation in training. Failure to present a project that at least has minimum functionality will lead to a review of your overall performance in the cohort between your trainer and the Manager of Technology that could result in a release.

1. Are there really no extensions?

A: There are not.